An Ensemble Deep Learning Model for Drug Abuse Detection in Sparse Twitter-sphere

Han Hu[*], Hai Phan[*], James Geller[*], Stephen Iezzi[*] and Sophie Di Lorio[†]
Ying Wu College of Computing
New Jersey Institute of Technology
Newark, U.S.
{hh255, phan, james.geller, sai3}@njit.edu[*];sophiediiorio@gmail.com[†]

Huy Vo
Dept. of Computer Science
The City College of New York
New York, U.S.
hvo@cs.ccny.cuny.edu

Xueqi Huang
Dept. of Computer Science
The City College of New York
New York, U.S.
xueqi.huang@nyu.edu

Thang Dinh
Dept. of Computer Science
Virginia Commonwealth University
Richmond, U.S.
tndinh@vcu.edu

Soon Ae Chun
Dept. of Computer Science
City University of New York
New York, U.S.
soon.chun@csi.cuny.edu

***Abstract*—As the problem of drug abuse intensifies in the U.S., many studies that primarily utilize social media data, such as postings on Twitter, to study abuse-related activities use machine learning as a powerful tool for text classification and filtering. However, given the wide range of topics of Twitter users, tweets related to drug abuse are rare in most of the datasets. This imbalanced data remains a major issue in building effective tweet classifiers, and is especially obvious for studies that include abuse-related slang terms. In this study, we approach this problem by designing an ensemble deep learning model that leverages both word-level and character-level features to classify abuse-related tweets. Experiments are reported on a Twitter dataset, where we can configure the percentages of the two classes (abuse vs. non-abuse) to simulate the data imbalance. Results show that our ensemble deep learning models exhibit better performance than ensembles of traditional machine learning models, especially on heavily imbalanced datasets.**

***Index Terms*—drug abuse, machine learning, deep learning, social media, public health**

I. Introduction

Misuse and abuse of prescription drugs and of illicit drugs have been major public health problems in the United States for decades. A "Public Health Emergency" declared in 2016 [1] and several official surveys [2], [3] all show that the problem has been getting worse in recent years. For example, the most recent reports from the National Survey on Drug Use and Health (NSDUH) [2] estimate that 10.6% of the total population of people ages 12 years and older (i.e., about 28.6 million people) have misused illicit drugs in 2016, which represents an increase of 0.5% over 2015. According to the Centers for Disease Control and Prevention (CDC), opioid drugs were involved in 42,249 known deaths in 2016 nationwide [3]. In addition, the number of heroin-related deaths has been increasing sharply over five years and has surpassed the number of firearm homicides in 2015 [4]. The emerging new problems, such as the epidemic of illicitly manufactured fentanyl (IMF) [5], marijuana-related traffic accidents [6], and marijuana use among adolescents [7] are posing further increasing threats to public health.

To fight this epidemic of drug abuse, methods of social media monitoring with wider scope and shorter response time are needed. Social media, such as Twitter, have been proven to be a sufficient and acceptably reliable data sources for social-level detection and monitoring tasks [8], [9]. Twitter is a popular social media platform that has 100 million daily active users and 500 million daily tweets (messages posted by Twitter users), [10] most of which are publicly accessible, on a wide range of topics.

We are using algorithms for filtering and classification for acquiring abuse-related tweets for analysis and monitoring. Filtering is the very first and most basic step toward extracting potentially useful tweets from the large number posted every day. Filtering, by itself, even with standard drug names (e.g. "heroin"), generally does not suffice to produce a dataset pure enough for practical use. Thus, machine learning classifiers have to be trained to further identify tweets that are related to drug abuse. However, most abuse-related Twitter datasets have the problem of imbalanced class distributions. Typical datasets, collected with only the names of drugs, may have 5% to 30% of positive (abuse-related) tweets, due to the topic diversity and language irregularity of tweets. The percentage of positive tweets decreases sharply when more keywords, especially slang names for drugs (e.g., "snow") and abuse behavior keywords (e.g., "snorting"), are included in a tweet dataset . The imbalanced class distribution and the noisy nature of the Twitter data make it hard to train a machine learning classifier with good performance.

In this paper, we propose an ensemble of two types of deep learning-based methods as better options, among classifiers, for situations in which the collected data is inevitably imbalanced, because they are more robust than traditional machine learning models. We combine word-level CNN models and character-level CNN models to perform classification. We compare our models with baseline models on a dataset we collected, where we can configure the class distribution of positive versus negative tweets in the training data and test data. By changing the percentage of positively and negatively labeled data in the dataset, we can simulate the imbalanced datasets that were collected by different means. We validate the performance of different models in a variety of settings to get a clearer picture of how imbalanced data affect classification performance. We also present our work on acquiring labeled tweets using the Amazon Mechanical Turk platform for lowering the cost and improving the efficiency of training data preparation. Finally, we demonstrate the results of word frequency analysis and temporal analysis of collected tweets, labeled by our proposed model.

The paper is organized as follows. In Section II, we discuss related work about the drug abuse problem, studies that utilize social media for it, and deep learning models for text classification of tweets. In Section III, we describe the methods we used to build and vary the dataset, and we define our models in detail. In Section IV, we report our experimental results. Sections V and VI conclude the paper.

II. Related Work

A. The Drug Abuse Problem

Large scale surveys, such as NSDUH [2], Monitoring the Future [11], the MedWatch program [12], and the results derived from these surveys [13], clearly show that there is an epidemic of drug abuse across the United States. However, a recent report [14] states that the estimated number of deaths due to prescription drugs could be inflated due to the difficulties in determining whether a drug is obtained by prescription or not. We assert that the ambiguities highlighted in this new report raise questions about the reliability of the earlier surveys, and thus, such a report illustrates the potential value of social media-based studies.

B. Related Studies

Chary et al. [15] performed semantic analysis on 3.6 million tweets with 5% labeled and found that they have a significant correlation of agreement with the NSDUH data. Hanson et al. [16] conducted a quantitative analysis on 213,633 tweets discussing “Adderall,” a prescription stimulant commonly abused among college students. Furthermore, Shutler et al. [17] performed a qualitative analysis of prescription opioid-related tweets and found that indications of abuse were common.

Debanjan et al. [18] performed a comprehensive study on using deep learning models to identify mentions of drug intake in tweets. Han et al. [19] showed the potential of applying deep learning models in a drug abuse monitoring system to detect abuse-related tweets. Sarker et al. [9] proposed a supervised classification model that aggregates several traditional machine learning models to classify drug abuse tweets and non-abuse tweets. Katsuki et al. [20] trained SVM on a dataset of 1,000 tweets for classification of tweets for relevance and favorability of online drug sales.

C. Deep Learning

Deep learning [21] has become popular in recent years. Deep learning based artificial neural network models have been applied to many NLP tasks with great success. Convolutional Neural Networks (CNN) are a branch of deep learning that can capture local correlations and extract global correlations from sequential data, which was found to perform well in NLP tasks such as text classification. Kim [22] demonstrates a basic model structure for utilizing CNN with pre-trained word embeddings (i.e., Word2vec) to perform sentence level NLP tasks. Zhang et al. [23] proposed to use char-level (character-level) features, instead of word-level features, with a deep CNN structure, and they achieved state-of-the-art performance on some large benchmark datasets. Kim et al. [24] proposed a char-level CNN structure that captures n-gram features of different sizes.

III. Methods

In this section, we present the definition of the *drug abuse-related risk behavior detection problem*, our methods for collecting tweets, our methods for labeling tweets, and our ensemble deep learning approach.

A. Problem Definition

In this paper, our first goal is to build a Twitter dataset consisting of tweets that are related to drug abuse risk behaviors (**positive** tweets), and tweets that are not (**negative** tweets). The “drugs” in the term “drug abuse risk behaviors” in this study include Schedule 1 and Schedule 2 drugs and their derivatives [25], including marijuana, heroin, cocaine, fentanyl, etc. The reasons that we include *marijuana* even though it is legalized in several states are that: **(1)** *Marijuana* is still a controlled substance on the federal level, whether for medical use or recreational use; and **(2)** Marijuana can still cause harm to adolescents [7], can cause “use disorder” [13], and is related to traffic fatalities [6]. The term “abuse risk behavior” can be defined as “The existence of likely abusive activities, consequences, and endorsements of drugs.” Tweets that contain links to or summarize news and reports related to drug abuse, and tweets that merely express opinions about drug abuse, are counted as negative in this study. Our main goal in this paper is to train a model that can accurately classify positive and negative tweets in a highly imbalanced (drug abuse) dataset.

B. Data Collection

Although there are human-labeled drug abuse Twitter datasets (e.g. Sarker’s dataset [9]) available, due to Twitter’s data policy, which prohibits the direct sharing of tweet contents, by the time we accessed the tweets in that dataset, more than 40% of tweets were either removed or hidden from the public. This significantly affects the quality and integrity of existing publicly available datasets. Therefore, we needed to build a new dataset from scratch.

In our framework, raw tweets were collected through a set of Application Programming Interfaces (Twitter APIs) via keyword filtering. By defining a set of keywords, the API will fetch tweets that contain any of the keywords from either the real-time stream of tweets or from archived tweets. For a more complete coverage of drug-related topics, we selected three types of keywords: **(1)** Full and official names of drugs, e.g. marijuana, cocaine, OxyContin, fentanyl, etc.; **(2)** Slang terms for drugs, e.g. pot, blunt, coke, crack, smack, etc.; and **(3)** Drug abuse-related behaviors and slang terms, e.g., high, amped, addicted, headache, dizzy, etc. The number of keywords we used was limited to 400 by the Twitter APIs. From Jan 2017 to Feb 2017, we collected 3,265,153 tweets in total.

C. Data Annotation

To annotate tweets (i.e., as positive or negative) from the dataset at low cost and high efficiency, we utilized the crowdsourcing platform Amazon Mechanical Turk (AMT). AMT is a well-known crowdsourcing platform where posters can post jobs and workers finish jobs for micro-payments. A literature study [26] evaluated AMT as a trustworthy platform to obtain labeled data. To increase the concentration of positive tweets in the labeled dataset, we first trained an SVM classifier as a pre-filter utilizing 1,794 labeled tweets from Han's study [19]. To ensure the quality of the AMT-labeled dataset, three members in our research team with experience in health informatics labeled each of the 1,794 tweets used for training the pre-filter SVM model, utilizing the same instructions used by AMT workers. An inter-annotator agreement score of 0.424, computed with Krippendorff's α, was achieved.

Next, we applied the trained SVM classifier to the collected tweets. Then 5,000 tweets that were labeled as positive by the SVM classifier (with high classification confidence; prediction probability > 0.8), were sampled to be posted on AMT. A comprehensive guide was provided to the workers explaining the labeling criteria and sample cases. The reward for each label was $0.05. Each tweet was labeled by three different annotators, and an inter-annotator agreement score of 0.456, indicated by Krippendorff's α, was achieved. There were 15 tweets omitted in the process due to labelling errors. The final labels of the 4,985 remaining tweets were determined by majority voting.

In a final step, an independent annotator went through tweets annotated by AMT and reported a rate of disagreements less than 5%. Given the comparable α scores and low disagreement rate, we consider the quality of the AMT labels to be as good as an average annotator group.

D. Feature Extraction

Machine learning models require numerical features to work with. Feature extraction transforms text features into numerical features in the form of vectors. To cover the content ambiguity in drug abuse-related tweets, a variety of feature extraction methods were used in this study. In our word-level CNN models, we used pre-trained word embedding models that were trained on large corpora to transform words into dense vectors. We tested several pre-trained models as Debanjan's work [18] suggested. With our word-level CNN model, the *Drug_chatter* embedding had the best average performance on our dataset; thus, it was chosen as the pre-trained word embedding model for this study. The details of the tested word embedding models are shown in Table I.

TABLE I
DETAILS OF PRE-TRAINED WORD EMBEDDINGS

Name	Model	Corpus	Vocabulary	Dim.
GoogleNews [27]	Word2vec	100 billion words	3 million	300
Glove_Comm [28]	Glove	42 billion words	1.9 million	300
Godin [29]	Word2vec	400 million tweets	3 million	400
Drug_chatter [30]	Word2vec	1 billion tweets	1.6 million	400

The preprocessing and tokenization procedure of raw tweets in our word-level CNN models is as follows: **(1)** The non-word characters, including HTML symbols, punctuation marks, and foreign characters in raw tweets were removed by regular expressions; **(2)** We used the *preprocessor package* [31] to tokenize the tweets. All characters were lower-cased. Special entities, including Emojis, URLs, mentions, and hashtags, were removed or replaced with special tokens; **(3)** Words with three or more repeating characters were reduced to at most three successive characters; and **(4)** Stop-words were removed according to a custom stop-word list derived from NLTK's stop-word list [32], where only pronouns and prepositions are kept as stop-words. Stemming was not applied, since many pre-trained Word2vec models did not use stemming when trained. Finally, each tweet was converted to a sequence of 400-dimensional vectors. Considering that the length limit of each tweet nowadays is 280 chars, the sequence length was set to 40. All-zero vectors were used as paddings for tweets that had fewer than 40 words. For out-of-vocabulary words, if such a word had more than two identifiable frequent adjacent (±2 adjacency) words in the dataset, it was represented as the average vector of these adjacent word vectors; otherwise, it was represented as the average vector of all infrequent (fewer than three occurrences in the dataset) word vectors.

In our char-level CNN, the preprocessing step only turns all characters to lower case as suggested by [23]. We used the following 70-character charset ('\n' is the new-line character): {abcdefghijklmnopqrstuvwxyz0123456789-,;.!?:'"/\|_@#$%^&*~`+-=<>()[]{}\n}. Each char in the charset was then converted into a 128-dimensional trainable random vector. Instead of being fixed, as the word embeddings, the character embeddings are trained along with other layers in the model.

We also replicated the features extracted in Sarker's study [9], including: **(1)** The tokenization process, where the usernames (e.g., @someone) and URLs were removed from raw tweets. The tweets were then lower-cased and stemmed with the Porter Stemmer. The same tokenizer and POS-tagger was used to tokenize the tweets and acquire POS tags for words; **(2)** The abuse-indicating term features, consisting of the presence and the counts of abuse-indicating terms obtained from [16]; **(3)** The drug-slang lexicon features, consisting of the presence and the counts of terms longer than five characters found in an online drug abuse dictionary [9]; **(4)** The word cluster features, represented by 150-dimensional one-hot vectors, were constructed by identifying words that belong to certain word clusters in the dataset [9] that contains 150 drug-related word clusters; and **(5)** The synonym expansion features, accomplished by identifying all synonyms of all nouns, verbs, and adjectives in the tokenized tweets using WordNet [33]. The vocabulary size of the synonym expansion features was set to 2,500, which means that only the top 2,500 frequent synonyms were included, and the vector length of this feature was also 2,500. 1-, 2- and 3-grams were identified from the processed tokens with the vocabulary size set to 5,000, which means that each tweet was transformed into a vector of 5,000 dimensions. An element in this vector is the count of an n-gram feature in the tweet. Finally, for each tweet, all of the features were concatenated to form a vector of the size 7,654 (5,000+2,500+2+2+150). The abuse-indicating term features, the drug-slang lexicon features, the synonym expansion features, and the word cluster features were also used as auxiliary features in our deep learning models.

E. An Ensemble Deep Learning Model for Drug Abuse Detection

In this section, we present our novel ensemble deep learning model for drug abuse detection by integrating extracted features from tweets into CNN models. In fact, both the word-level CNN and the char-level CNN share a similar structure as shown in Fig. 1

The inputs of our word-level CNN are vectors of shape [40, 400] where 40 is the maximum sequence length (number of words allowed) in an input tweet, and 400 is the length of the pre-trained word embeddings. The input of our char-level CNN is shaped as [280, 128] where 280 is the maximum possible length of a tweet, and 128 is the length of the vector representation of each character in the charset. The auxiliary features in the input include: **(1)** The synonym expansion features in the form of synonymous words were directly concatenated with the input tweets (before they were transformed into vectors); and **(2)** The remaining auxiliary features, in the form of 154-dimensional vectors, were concatenated to the last hidden layer of the fully-connected layers.

For each convolution kernel size, the word-level CNN model has two convolution layers with ReLU activation functions stacked together. Each is followed by a max-pooling layer. The char-level CNN model has one convolution layer for each convolution kernel size with $tanh$ activation function, followed by a global-max-pooling layer that performs max-pooling over all outputs of convolution layers with different kernel sizes. Both models have one fully connected layer component, consisting of two fully connected layers with 1024 hidden units each, and one Softmax output layer with two units. The activation functions are slightly different, as the word-level CNN model uses ReLU, while the char-level CNN model uses SELU. The output of the last hidden layer is concatenated with vectors of abuse-indicating term features, drug-slang lexicon features, and word cluster features, before being fed into the output layer.

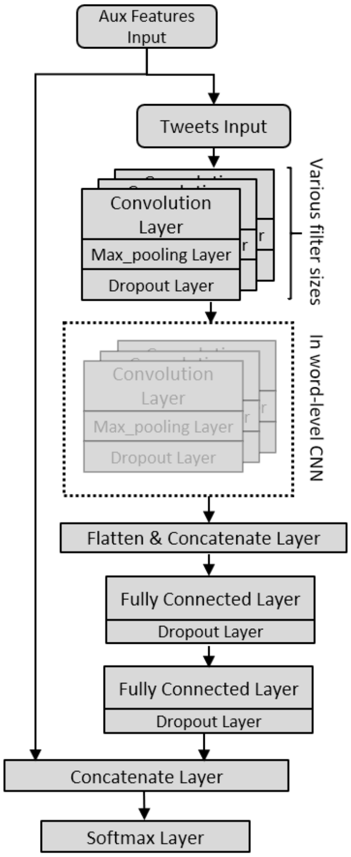

Fig. 1. CNN model structures

Both models use the Adam optimizer with an initial learning rate of 0.0005, and a decay rate of 0.02. Dropout is used after each convolution layer in our word-level CNN model, and after each fully connected layer for both models. Class weight is set to be inversely proportional to class frequencies. Batch size is set to 64. The deep learning models are implemented with *Keras* [34]. The parameter settings for each layer are shown in Table II.

TABLE II
CNN LAYER PARAMETERS

Model	Parameter	Value
SVM	C	{0.5, 1.0}
	gamma	{0.01, 0.5, 0.1}
Random Forest	#_trees	500
	max_depth	20
	max_features	0.4
	max_leaf_nodes	50
Naïve Bayes	distribution	multinomial

Model ensembles (stacking) were also used in Sarker's study [9] to reduce variability and bias, in order to improve prediction performance. An ensemble model takes the outputs of multiple prediction models to train a meta-learner that gives the final predictions. In this study, we use majority voting as the meta-learner. We apply the same ensemble strategy to both our deep learning models and the baseline models.

F. Baseline Models

We choose three well-known machine learning models, namely, SVM, Random Forest, and Naïve Bayes, that were used in Sarker's study [9], as baseline models. Direct comparisons of performance will be reported, between our models and those models. We used the Scikit-learn [35] Python library for all three models. All baseline models are fed with the same set of features as aforementioned. Class weight, if applicable, is set to be inversely proportional to class frequency. We tuned each model in our dataset for best performance. The parameter settings of baseline models are shown in Table III.

TABLE III
MACHINE LEARNING MODELS PARAMETERS

Layer	**Parameter**	**Word-level CNN**	**Char-level CNN**
Input	size	40*400	280*128
	kernel_width	{2,3,4,5}	{3,5,7,10}
Convolution	#_kernels	128 for each kernel width	256 for each kernel width
	pool_size	2	-
	activation function	Relu	Tanh
Dropout	dropout_rate	0.1	0.1
Fully-connected (with dropout)	#_hidden_units	{1024, 1024}	{1024, 1024}
	dropout_rate	0.5	0.5
	activation function	Relu	Selu
Softmax layer	size	2	2

IV. EXPERIMENTS

A. Dataset

Before the dataset was used in our experiments, we removed duplicate tweets from it. Our final labeled dataset contains 4,736 tweets with 2,657 positive labels and 2,079 negative labels. To simulate the data imbalance scenarios, we configured the class distribution and pre-sampled the dataset into six blocks for each distribution scenario, for 6-fold cross-validation. Each model was trained and tested on the same sets of training and test data to ensure a fair comparison. The number of data points included in each distribution scenario was maximized, but it was inevitably different between scenarios. Table IV shows the dataset in each class distribution scenario.

TABLE IV
DATASET VARIANTS

Class Distribution (positive: negative)	**# of training data**	**# of testing data**
50:50 split	3450	690
40:60 split	2850	570
30:70 split	2450	490
20:80 split	2150	430
10:90 split	1900	380

B. Performance Measures

We used the standard performance measures, i.e., accuracy, precision, recall, and F1-score, in our experiments. Table V shows how these measures are defined. We report precision, recall, and F1-score separately for negative labels by treating them as the "positive" labels in formulas. The results of negative labels can better show whether the trained model is biased or not. However, the measures for positive (drug abuse-related) labels are the main criteria when determining whether one model is better than the others. The F1-score was regarded as the main performance criterion. When two models have similar F1-scores, the model with the smaller difference between precision and recall would be chosen as the better model. Also, note that the accuracy should only be treated as an auxiliary measure, due to the imbalanced class distribution.

TABLE V
DEFINITIONS OF MEASURES

Measure	**Formula**	**Keys**
Accuracy	$\frac{tp+tn}{tp+fn+fp+tn}$	tp: # true positives tn: # true negatives fp: # false positives fn: # false negatives
Precision	$\frac{tp}{tp+fp}$	
Recall	$\frac{tp}{tp+fn}$	
F1-score	$\frac{2*tp}{2*tp+fn+fp}$	

C. Experiment Dessign

Our main objective in this experiment is to directly compare the performances of the ensemble traditional machine learning model and the ensemble deep learning model. For the ensemble traditional machine learning model, two of each type of baseline models, six in total, were trained and ensembled together. For the ensemble deep learning model, six models of three types (two for each type) were used. The three types are denoted as follows. **(1)** "char-aux" is the char-level CNN model with auxiliary features. **(2)** "char-cnn" is the plain char-level CNN without any auxiliary features. **(3)** "word-aux" is the word-level CNN model with all auxiliary features. For deep learning models, it is extremely easy to overfit, due to the rather small number of training and test data elements; thus, the model is saved at each training epoch, and the best epoch is found among the saved models. For each class distribution scenario, each model is trained with the same six sets of training data and tested on the corresponding test data. All results reported are averaged results from the 6-fold cross-validation.

D. Results

Table VI shows the results for all individual models and two ensemble models. The ensemble model results are separated from the individual models for easier viewing. The highest value of each measure is marked in bold font.

There is an interesting trend in the results of ensemble models. When the data is balanced or nearly balanced, the traditional ensemble machine learning model has a better performance than the ensemble deep learning model. At 50:50 and 40:60 splits, the ensemble machine learning model is

TABLE VI
DATASET VARIANTS

Class Distribution: 50:50 split								
Measure	**char_expan**	**char_cnn**	**word_expan**	**SVM**	**Random Forest**	**Naïve Bayes**	**Ensemble CNN**	**Ensemble ML**
Accuracy	0.8506	0.8477	0.8466	0.8415	**0.8586**	0.8384	0.851	**0.8575**
Precision_p	0.8315	0.824	0.8198	0.8063	**0.8404**	0.8319	**0.8468**	0.835
Precision_n	0.8723	0.8754	**0.8795**	0.887	**0.8795**	0.8462	0.8556	**0.884**
Recall_p	0.8797	0.8845	0.8894	**0.9**	0.886	0.8493	0.8575	**0.8918**
Recall_n	0.8215	0.8109	0.8039	0.7831	**0.8312**	0.8275	**0.8444**	0.8232
F1_score_p	0.8549	0.8531	0.8529	0.8504	**0.8624**	0.8402	0.852	**0.8623**
F1_score_n	0.8461	0.8418	0.8396	0.8315	**0.8544**	0.8365	0.8499	**0.8523**
Class Distribution: 40:60 split								
Measure	**char_expan**	**char_cnn**	**word_expan**	**SVM**	**Random Forest**	**Naïve Bayes**	**Ensemble CNN**	**Ensemble ML**
Accuracy	0.8528	**0.8563**	0.843	0.8444	0.8494	0.8427	0.8567	**0.8582**
Precision_p	0.8007	0.8055	0.7818	**0.8104**	0.777	0.7862	**0.8079**	0.8047
Precision_n	0.8911	0.8934	0.8909	0.8669	**0.9089**	0.8848	0.8921	**0.898**
Recall_p	0.8421	0.8454	0.8443	0.7982	**0.8746**	0.8341	0.8428	**0.8531**
Recall_n	0.8599	0.8635	0.8421	**0.8752**	0.8326	0.8484	**0.866**	0.8616
F1_score_p	0.8207	**0.8248**	0.8113	0.8041	0.8229	0.8093	0.8249	**0.828**
F1_score_n	0.8751	**0.8781**	0.8654	0.871	0.869	0.8661	0.8788	**0.8793**
Class Distribution: 30:70 split								
Measure	**char_expan**	**char_cnn**	**word_expan**	**SVM**	**Random Forest**	**Naïve Bayes**	**Ensemble CNN**	**Ensemble ML**
Accuracy	0.8522	0.8507	0.8483	0.8429	**0.8537**	0.8452	**0.8599**	0.8595
Precision_p	0.7253	0.7223	0.718	**0.7467**	0.7137	0.7218	0.7402	**0.7426**
Precision_n	0.9186	0.9175	0.9162	0.8834	**0.9337**	0.9072	**0.9203**	0.918
Recall_p	0.8209	0.818	0.8158	0.7234	**0.8583**	0.7914	**0.8231**	0.8163
Recall_n	0.8656	0.8647	0.8622	**0.8941**	0.8518	0.8683	0.8756	**0.878**
F1_score_p	0.7695	0.7666	0.7635	0.7336	**0.7789**	0.7538	**0.7792**	0.7771
F1_score_n	**0.8912**	0.8902	0.8883	0.8884	0.8907	0.887	0.8973	**0.8974**
Class Distribution: 20:80 split								
Measure	**char_expan**	**char_cnn**	**word_expan**	**SVM**	**Random Forest**	**Naïve Bayes**	**Ensemble CNN**	**Ensemble ML**
Accuracy	**0.8624**	0.8568	0.8506	0.8384	0.8475	0.8527	**0.8674**	0.8508
Precision_p	**0.6325**	0.6128	0.5965	0.564	0.5838	0.6261	**0.6416**	0.5908
Precision_n	0.9358	0.9427	0.9463	**0.9583**	0.9525	0.914	0.9397	**0.9526**
Recall_p	0.7558	0.7868	0.8023	**0.8547**	0.8295	0.6609	0.7713	**0.8295**
Recall_n	0.8891	0.8743	0.8626	0.8343	0.852	**0.9007**	**0.8915**	0.8561
F1_score_p	**0.6878**	**0.6878**	0.6823	0.6792	0.685	0.6425	**0.7001**	0.69
F1_score_n	**0.9117**	0.907	0.9022	0.8919	0.8994	0.9072	**0.9149**	0.9017
Class Distribution: 10:90 split								
Measure	**char_expan**	**char_cnn**	**word_expan**	**SVM**	**Random Forest**	**Naïve Bayes**	**Ensemble CNN**	**Ensemble ML**
Accuracy	0.8638	0.8664	0.8445	0.8355	0.8592	**0.8961**	**0.8728**	0.8636
Precision_p	0.4112	0.4153	0.376	0.3609	0.3875	**0.4762**	**0.4338**	0.3975
Precision_n	0.9678	0.9677	0.9647	**0.9755**	0.9608	0.9247	**0.9672**	0.9607
Recall_p	0.7368	0.7346	0.7171	**0.8114**	0.6776	0.2939	**0.7281**	0.6754
Recall_n	0.8779	0.8811	0.8587	0.8382	0.8794	**0.963**	**0.8889**	0.8845
F1_score_p	0.5243	**0.5275**	0.4882	0.499	0.4925	0.3611	**0.5389**	0.4999
F1_score_n	0.9204	0.9221	0.9079	0.9015	0.9182	**0.9434**	**0.9261**	0.921

superior over the ensemble deep learning model for most of the criteria. We argue that this is partially due to the relatively small dataset size. When the data becomes more imbalanced, e.g., at a 30:70 split, the ensemble deep learning model becomes better and has a higher F1-score for positive labels and virtually the same F1-score for negative labels, compared with the traditional ensemble machine learning model. At 20:80 and 10:90 splits, the ensemble deep learning model takes the lead, most significantly in each measure for positive labels. We argue that the larger model capacity and the ability of the deep learning models to learn more complex non-linear functions can better distinguish the semantic differences between positive tweets and negative tweets, when the distribution of classes is heavily imbalanced.

Looking at individual machine learning models, Random Forest and SVM show a strong performance on all datasets, and they are especially good when the dataset is balanced. Naïve Bayes also has a good performance on a balanced dataset, but on an imbalanced dataset, it is heavily biased towards negative labels and has a poor performance for positive labels. Deep learning models generally have more stable performance, compared to traditional machine learning models, across all datasets, and a smaller difference between precision and recall, but their peak performances are not as good. Comparing between deep learning models, auxiliary features do not give char-level CNN significant performance boost, and word-level CNN is also not as good as the char-level CNN model. However, in additional results that are not

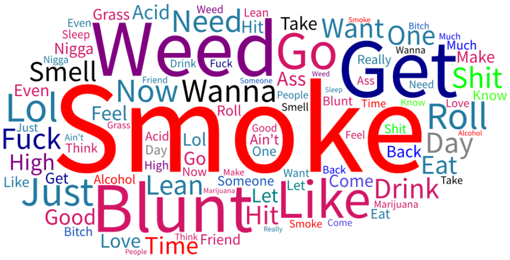

Fig. 2. Word cloud of positive tweets.

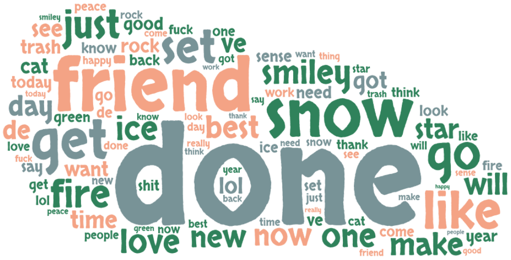

Fig. 3. Word cloud of negative tweets.

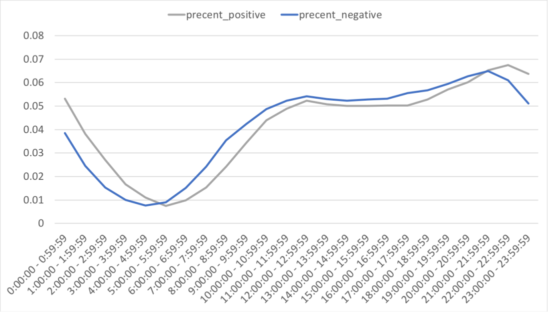

Fig. 4. Time-of-the-day distribution of tweets.

shown in this paper due to space limitations, auxiliary features give the plain word-level CNN model a performance boost.

In investigating the performance of each individual model and the ensemble model that includes it, we can see that our ensemble strategy works well for deep learning models, as most of the measures for the ensemble model are higher than for any of its components' corresponding measures. This effect was only observed a few times for traditional machine learning models. We expect that, by using more complicated ensemble strategies, deep learning has the potential to reach an even better performance level.

E. Sample Analysis

To gain insights into drug abuse risk behaviors mentioned on Twitter, we analyzed over 3 million drug abuse-related tweets. These tweets were labeled by our ensemble deep learning model. There are 117,326 tweets classified as positive, and 3,077,827 tweets classified as negative. The positive tweets correspond to 3.67% of the whole dataset. The word distribution in positive tweets (Fig. 2) is remarkably different from the word distribution in negative tweets (Fig. 3). In fact, abuse-indicating tweets usually consist of abuse-indicating terms, and drug names, such as "blunt," "high," "smoke," "weed," "marijuana," "grass," "juic," etc. In addition, the high concentration of swear words, e.g., "s**t," "f**k," "as*," "bit**," etc., clearly suggests the expression patterns that may be popular among drug abusers. These expression patterns are less likely to exist in negative tweets.

We also extracted the local time of posting for each tweet and discovered some interesting patterns. The most significant and interesting pattern is shown in Fig. 4. The time-of-day distribution of positively and negatively labeled tweets are shown in light-colored (positive) and dark-colored (negative) lines. The x-axis shows the time of day in one-hour intervals, and the y-axis measures the percentage of tweets. There is a clear difference between the two lines, suggesting that abuse-indicating tweets may have a different temporal pattern from non-abuse tweets. This finding also supports the effectiveness of our approach. Presumably there is less drug use during working hours (extended in each direction, possibly due to commuting time).

V. Discussion

A. Model Choice

The word-level CNN is a more "conventional" type of model that utilizes the semantic features embedded in the pre-trained word embeddings. It is widely used in many studies and has a rather good performance. In our study, however, char-level CNN models are superior. The biggest reason for this might be the irregularity of Twitter language. The informal patterns of expression, wide topic range, and the ever-changing vocabulary (due to new or newly named drugs) make pre-trained word embeddings less effective on different datasets. In the drug domain, char-level CNN appears to work well on Twitter data, even on a moderately sized dataset and with the randomly initialized charset-embeddings.

B. Performance and Overfitting

One observation we made during model tuning is that our deep learning models, especially the word-level CNN model, are prone to overfitting. Even when using dropout, some models begin overfitting at as early as two epochs. The char-level CNN model has less of this problem. We view this as another argument against using pre-trained word embeddings on small-sized Twitter datasets, as the small number of short tweets cannot provide enough variability for the model to learn complex correlations.

C. Future Work

We have three future goals when utilizing social media for drug abuse detection and monitoring: **(1)** Build a bigger and more generalizable labeled dataset; **(2)** Utilize additional features, such as user metadata and timelines, and additional sources of slang terms, such as the Urban Dictionary; and **(3)** Build a real-time system that can continually monitor and study the patterns of drug abuse-related posts, not only on Twitter, but on other popular social platforms, e.g., Reddit.

VI. Conclusion

In this study, we investigated how the data imbalance issue influences the performance of classifiers that are trained for identifying tweets that are related to drug abuse. We first collected a dataset with a broad selection of drug abuse-related keywords and slang terms. We explored the use of the Amazon Mechanical Turk platform as a reliable source for acquiring human-labeled tweets, and we obtained a solid dataset. We designed an ensemble deep learning classification model with both word-level and char-level CNNs, and we conducted a direct comparison with traditional machine learning models on our dataset, with simulated class imbalance. Experimental results show that our ensemble deep learning models have better performance than traditional machine learning models when the data is off-balance. Results also show that the ensemble strategy we used is effective for improving deep learning models. Finally, our analysis of the collected three million tweets, labeled by our model, shows an interesting temporal pattern that agrees with our intuition.

References

[1] HHS Press Office. (2017, Oct.) Hhs acting secretary declares public health emergency to address national opioid crisis. [Online]. Available: https://www.hhs.gov/about/news/2017/10/26/hhs-acting-secretary-declares-public-health-emergency-address-national-opioid-crisis.html

[2] The Substance Abuse and Mental Health Data Archive. (2018, Jul.) National survey on drug use and health. [Online]. Available: https://www.samhsa.gov/data/data-we-collect/nsduh-national-survey-drug-use-and-health

[3] National Institute on Drug Abuse. (2018, Aug.) Overdose death rates. [Online]. Available: https://www.drugabuse.gov/related-topics/trends-statistics/overdose-death-rates

[4] Gun Violence Archive. (2018, Aug.) 2015 gun violence archive. [Online]. Available: http://www.gunviolencearchive.org/past-tolls

[5] J. K. O'Donnell, J. Halpin, C. L. Mattson, B. A. Goldberger, and R. M. Gladden, "Deaths involving fentanyl, fentanyl analogs, and u-47700-10 states, july-december 2016." *MMWR Morb Mortal Wkly Rep*, vol. 66, no. 43, pp. 1197–1202, Nov. 2017.

[6] B. Hansen, K. S. Miller, and C. Weber, "Early evidence on recreational marijuana legalization and traffic fatalities," Nat'l Bu. of Econ. Res., Working Paper 24417, Mar. 2018.

[7] E. J. D'Amico, A. Rodriguez, J. S. Tucker, E. R. Pedersen, and R. A. Shih, "Planting the seed for marijuana use: Changes in exposure to medical marijuana advertising and subsequent adolescent marijuana use, cognitions, and consequences over seven years," *Drug Alcohol Depend*, vol. 188, pp. 385 – 391, 2018.

[8] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. WWW*, ser. WWW '10, 2010, pp. 851–860.

[9] A. Sarker *et al.*, "Social media mining for toxicovigilance: automatic monitoring of prescription medication abuse from twitter," *Drug Saf*, vol. 39, no. 3, pp. 231–240, 2016.

[10] S. Aslam. (2018, Jan.) Twitter by the numbers. [Online]. Available: https://www.omnicoreagency.com/twitter-statistics/

[11] A. Arbor. (2018, Aug.) Monitoring the future. [Online]. Available: http://www.monitoringthefuture.org/

[12] US FDA. (2018, Aug.) Medwatch: The fda safety information and adverse event reporting program. [Online]. Available: https://www.fda.gov/safety/medwatch/

[13] D. S. Hasin *et al.*, "Us adult illicit cannabis use, cannabis use disorder, and medical marijuana laws: 1991-1992 to 2012-2013," *Jama Psychiatry*, vol. 74, no. 6, pp. 579–588, 2017.

[14] P. Seth, R. A. Rudd, R. K. Noonan, and T. M. Haegerich, "Quantifying the epidemic of prescription opioid overdose deaths," *Am J Public Health Res*, vol. 108, no. 4, pp. 500–502, 2018.

[15] M. Chary, N. Genes, C. Giraud-Carrier, C. L. Hanson, L. S. Nelson, and A. F. Manini, "Epidemiology from tweets: Estimating misuse of prescription opioids in the usa from social media," *J Med Toxicol*, vol. 13, no. 4, pp. 278–286, Dec. 2017.

[16] C. L. Hanson, S. H. Burton, C. Giraud-Carrier, J. H. West, M. D. Barnes, and B. Hansen, "Tweaking and tweeting: exploring twitter for nonmedical use of a psychostimulant drug (adderall) among college students," *J Med Internet Res*, vol. 15, no. 4, 2013.

[17] L. Shutler, L. S. Nelson, I. Portelli, C. Blachford, and J. Perrone, "Drug use in the twittersphere: a qualitative contextual analysis of tweets about prescription drugs," *J Addict Dis*, vol. 34, no. 4, pp. 303–310, 2015.

[18] D. Mahata, J. Friedrichs, Hitkul, and R. R. Shah, "#phramacovigilance - exploring deep learning techniques for identifying mentions of medication intake from twitter," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1805.06375

[19] H. Hu *et al.*, "Deep learning model for classifying drug abuse risk behavior in tweets," in *2018 IEEE Int. Conf. Healthcare Inform. (ICHI)*. IEEE, 2018, pp. 386–387.

[20] T. Katsuki, T. K. Mackey, and R. Cuomo, "Establishing a link between prescription drug abuse and illicit online pharmacies: analysis of twitter data," *J Med Internet Res*, vol. 17, no. 12, 2015.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, 2016, vol. 1.

[22] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1408.5882

[23] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1509.01626

[24] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1508.06615

[25] Office of the Law Revision Counsel, "Drug abuse prevention and control. definitions. 21 u.s.c sect. 802," 2018.

[26] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspect Psychol Sci*, vol. 6, no. 1, pp. 3–5, 2011.

[27] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th NIPS Vol.2*, ser. NIPS'13, 2013, pp. 3111–3119.

[28] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. EMNLP'14*, 2014, pp. 1532–1543.

[29] F. Godin. (2015) Twitter word2vec model. [Online]. Available: https://www.fredericgodin.com/software/

[30] A. Sarker and G. Gonzalez, "A corpus for mining drug-related knowledge from twitter chatter: Language models and their utilities," *Data Brief*, vol. 10, pp. 122 – 131, 2017.

[31] S. Özcan, "Elegant and easy tweet preprocessing in python," "https://github.com/s/preprocessor", 2015.

[32] S. Bird and E. Loper, "Nltk: the natural language toolkit," in *Proc. ACL'04 Interact. Poster Demo. Sess.* Association for Computational Linguistics, 2004, p. 31.

[33] G. A. Miller, "Wordnet: a lexical database for english," *Commun ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[34] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[35] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J Mach Learn Res*, vol. 12, pp. 2825–2830, 2011.